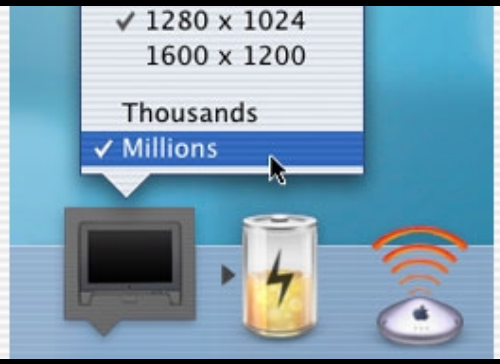
Session 604

Introduction to Enterprise Objects Framework



James Dempsey
Curriculum Manager
Apple iServices Training

Introduction

- An introduction to the Enterprise Objects Framework (EOF)

What You'll Learn

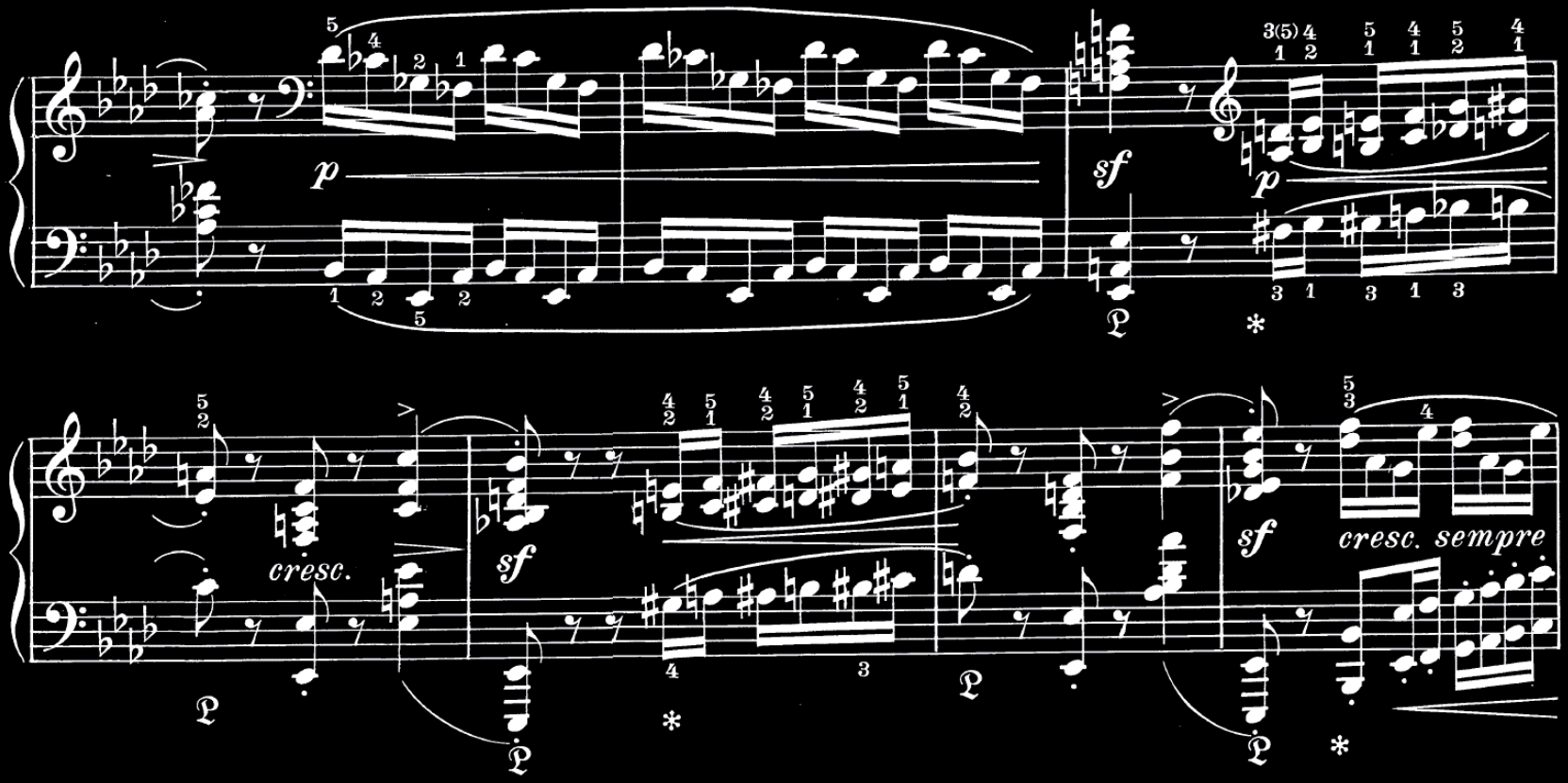
- An overview of the purpose of EOF
- A conceptual model of how EOF works
- Practical code for basic database CRUD
  - Create
  - Retrieve
  - Update
  - Delete

Technology Framework

- An introduction to the EOAccess and EOControl frameworks of EOF
- These frameworks form the core of database access in:
  - HTML-based WebObjects applications
  - Cocoa applications
  - Java Client applications

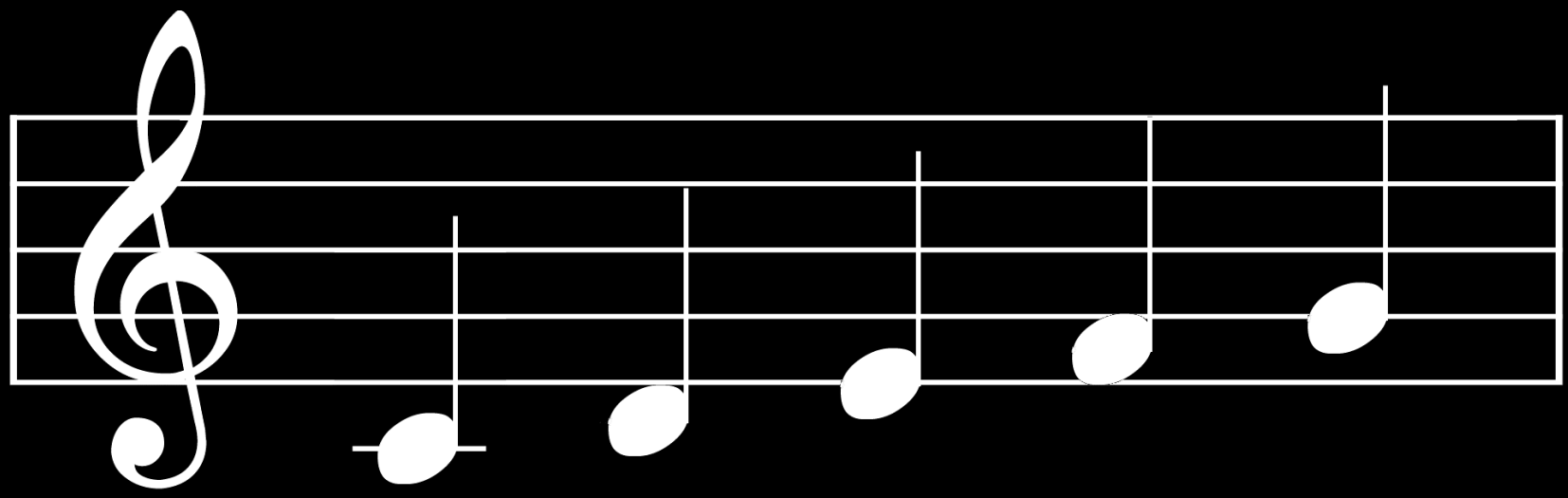
Solving Sophisticated Problems

Sophistication requires expertise

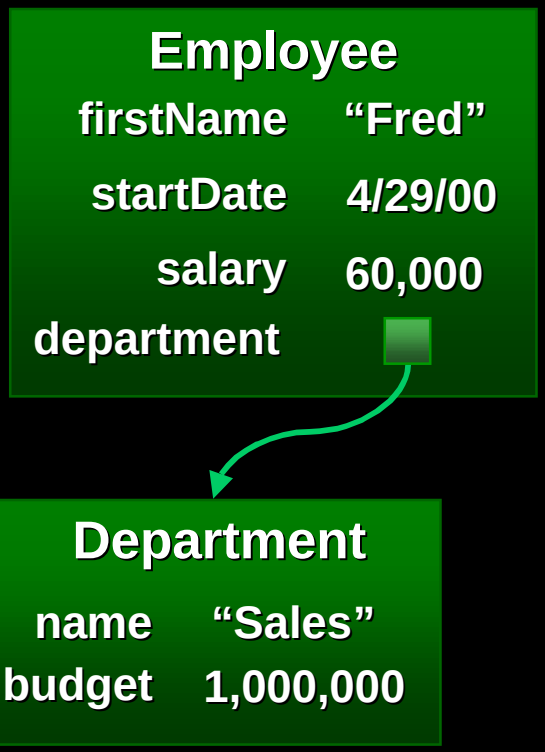


Fundamentals Are Key

Sophistication builds on fundamentals



From the Object Perspective



- Different classes of objects to model different types
- Each object stores data
- Each class of object has behavior—business logic

Object Persistence

Where do objects sleep at night?

- Store the data from objects when not in memory
- Archiving or serializing to a flat file works for small numbers of objects
- Need a high performance way of storing and retrieving large amounts of information
- Relational database seems a natural solution

Entities and Attributes

Employee

firstName	"Fred"
startDate	4/29/00
salary	60,000
department	

Employee

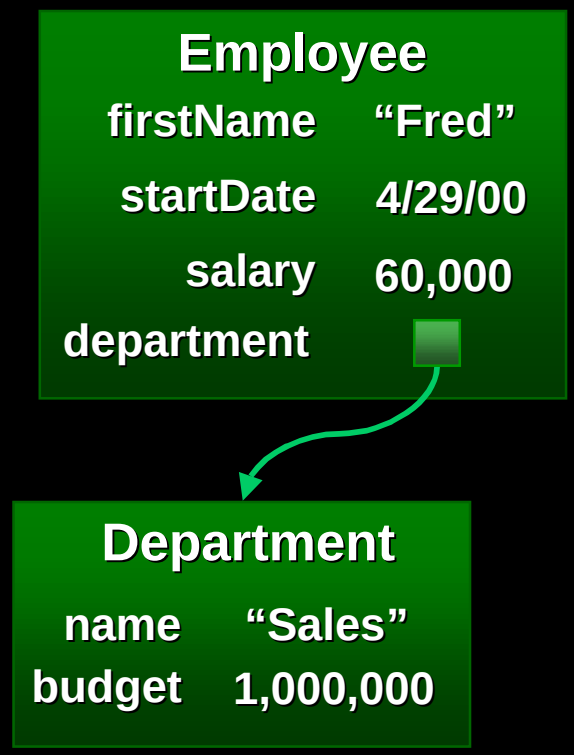
firstName	"Beth"
startDate	1/19/00
salary	65,000
department	

EMPLOYEE

EMP_ID	NAME	SALARY	ST_DATE	DEPT_ID
101	Fred	60,000	4/29/00	503
102	Beth	65,000	1/19/00	501

Relationship by Reference

A unique reference to another object



Relationship by Join

A unique primary key value in another table

EMPLOYEE

EMP_ID	NAME	SALARY	ST_DATE	DEPT_ID
101	Fred	60,000	4/29/00	503
102	Beth	65,000	1/19/00	501

DEPARTMENT

DEPT_ID	NAME	BUDGET
501	R&D	4,000,000
502	Sales	1,200,000

What are Enterprise Objects?

Affectionately known as EOs

Employee	
firstName	"Fred"
startDate	4/29/00
salary	60,000
department	

- Implement the EOEnterpriseObject interface
- Typically subclass EOGenericRecord
- Objects with built-in persistence

From Row to EO and Back

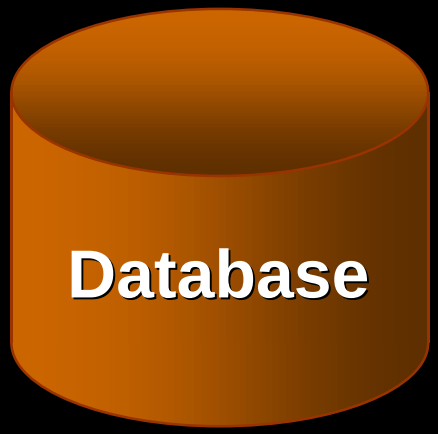
Employee

firstName	"Fred"
startDate	4/29/00
salary	60,000
department	

EMPLOYEE

EMP_ID	NAME	SALARY	ST_DATE	DEPT_ID
101	Fred	60,000	4/29/00	503

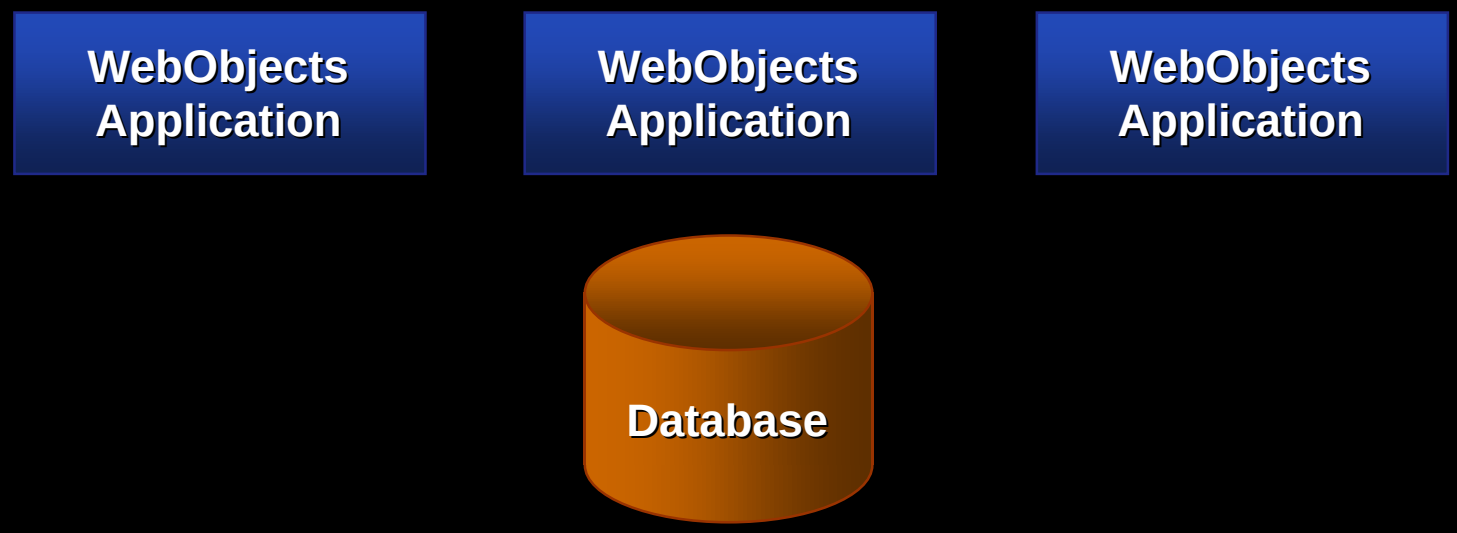
From the Database Perspective



- Data stored in database
- Logic can be executed in database
- EOF adds value to the power of a database

Distributed Processing

Off-load logic processing to application servers



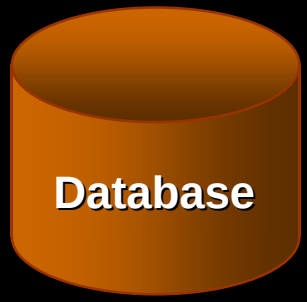
Logic Written in Standard Java

Object-oriented design in a widely used language

WebObjects Application

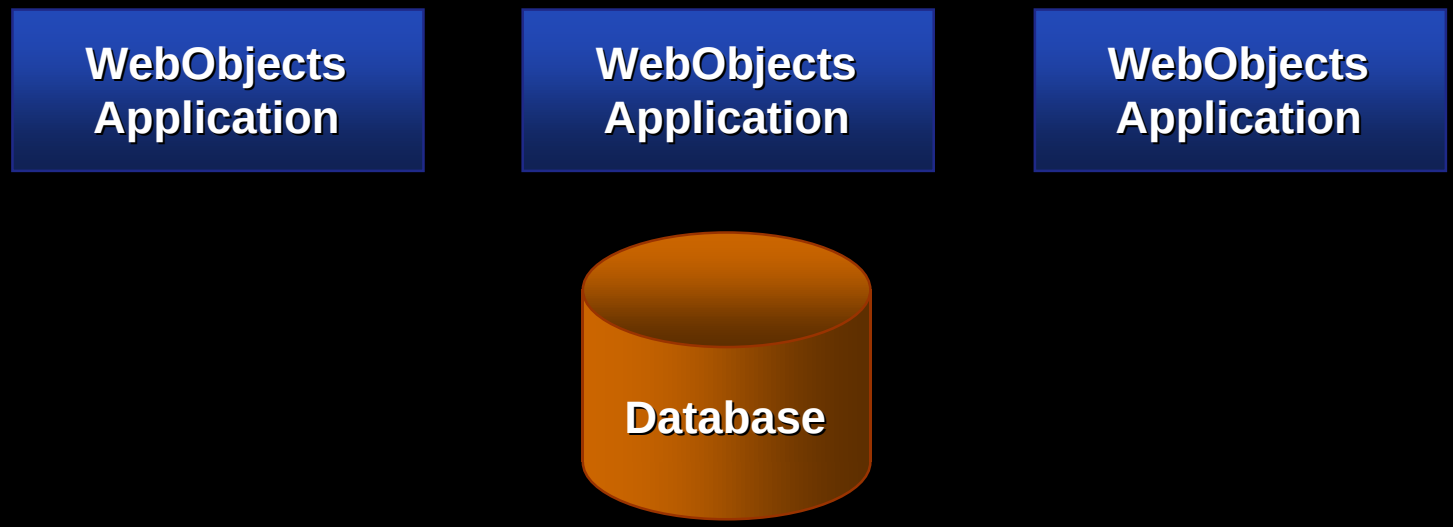
WebObjects Application

WebObjects Application



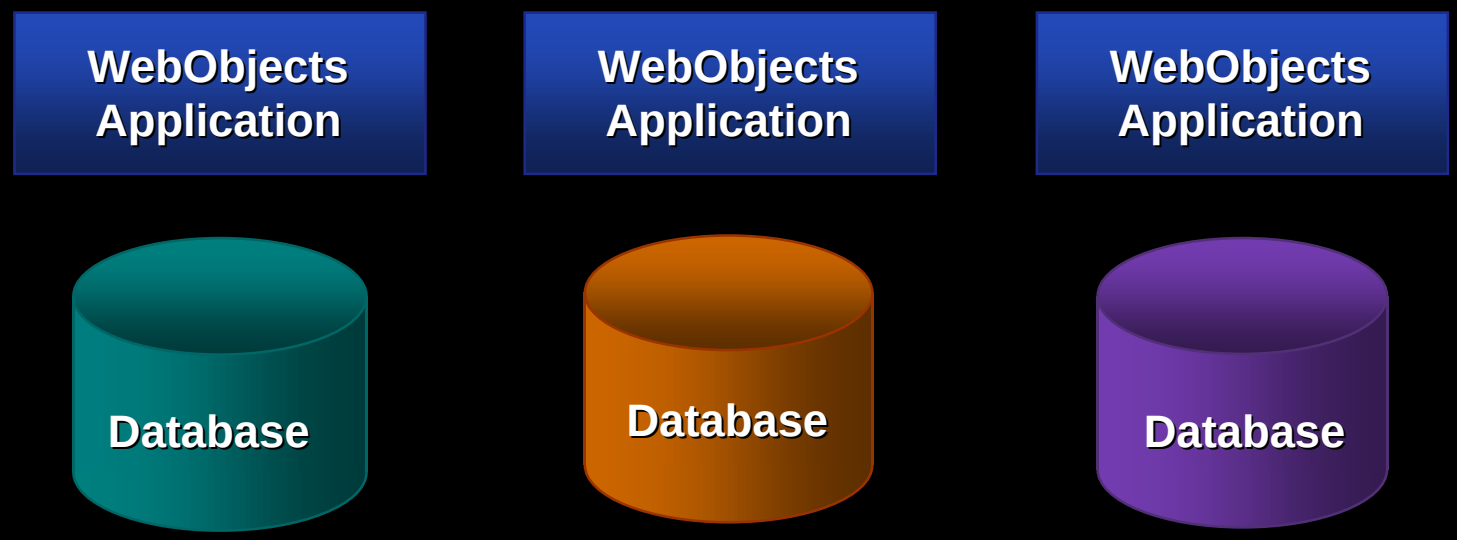
Dynamically Generated SQL

Transactions and statements generated by EOF



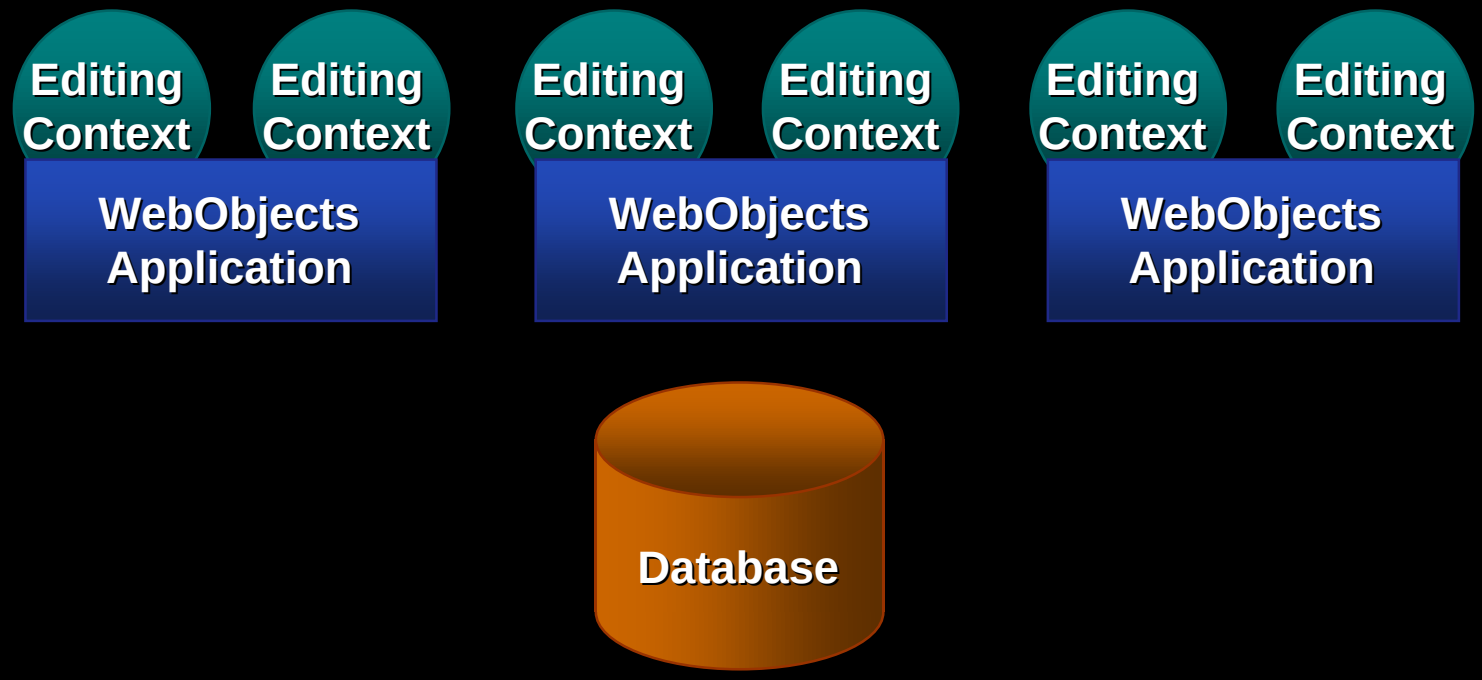
Multiple Data Sources

Using data from different sources abstracted at a low-level—programmatic interface remains the same



Separate ‘Scratch Pad’ Per User

Each user provided a separate context for making edits and canceling changes



EOEditingContext

The class you use most for programmatic control

- Fetches from the database
- Tracks updates to all EOs it has fetched
- Tracks inserts and deletes
- Stores edits until changes are saved or reverted

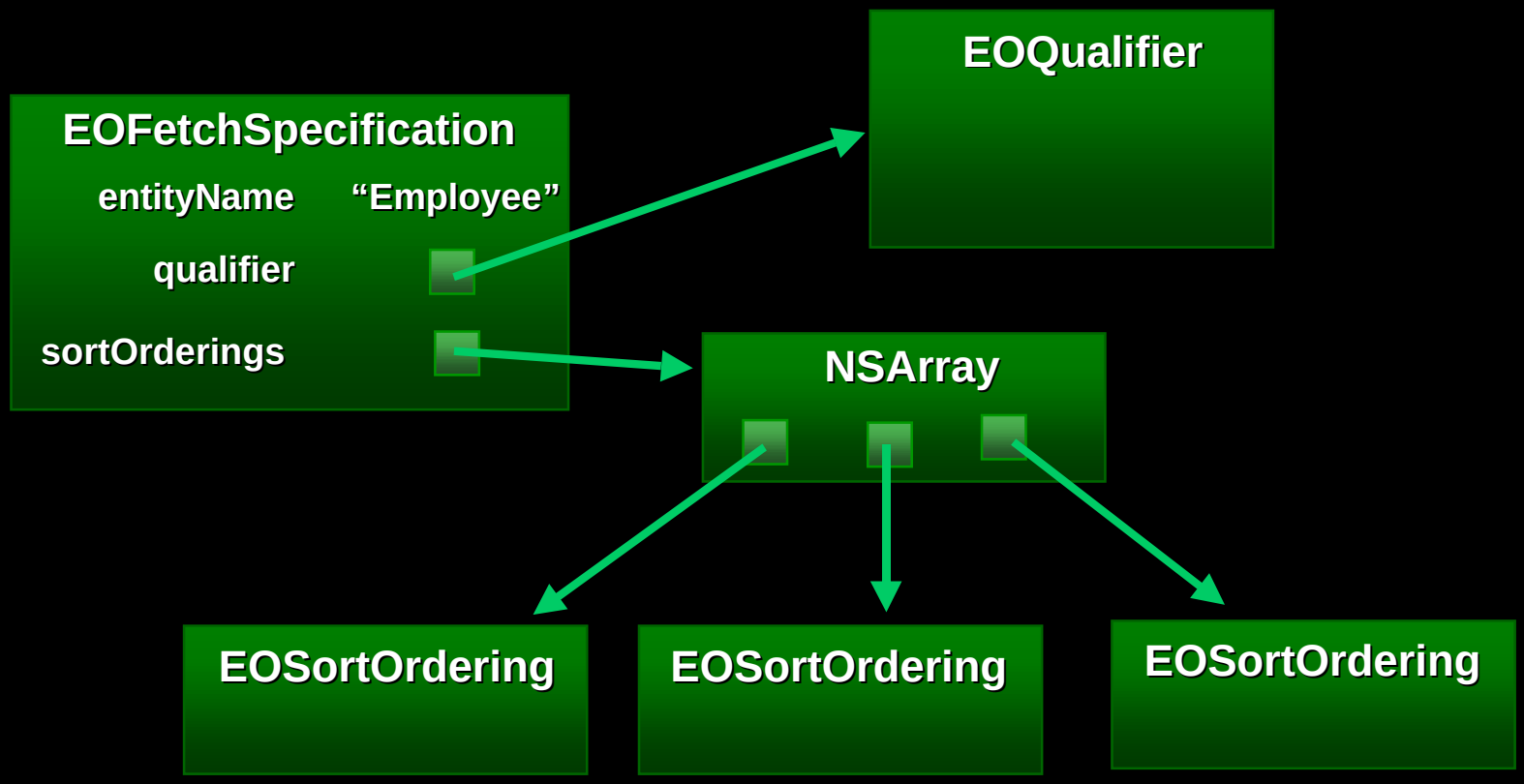
Fetching

Many techniques—all with same underlying steps

- Create a fetch specification
- Access an editing context
- Tell editing context to fetch using the specification
- Receive an array of retrieved Enterprise Objects

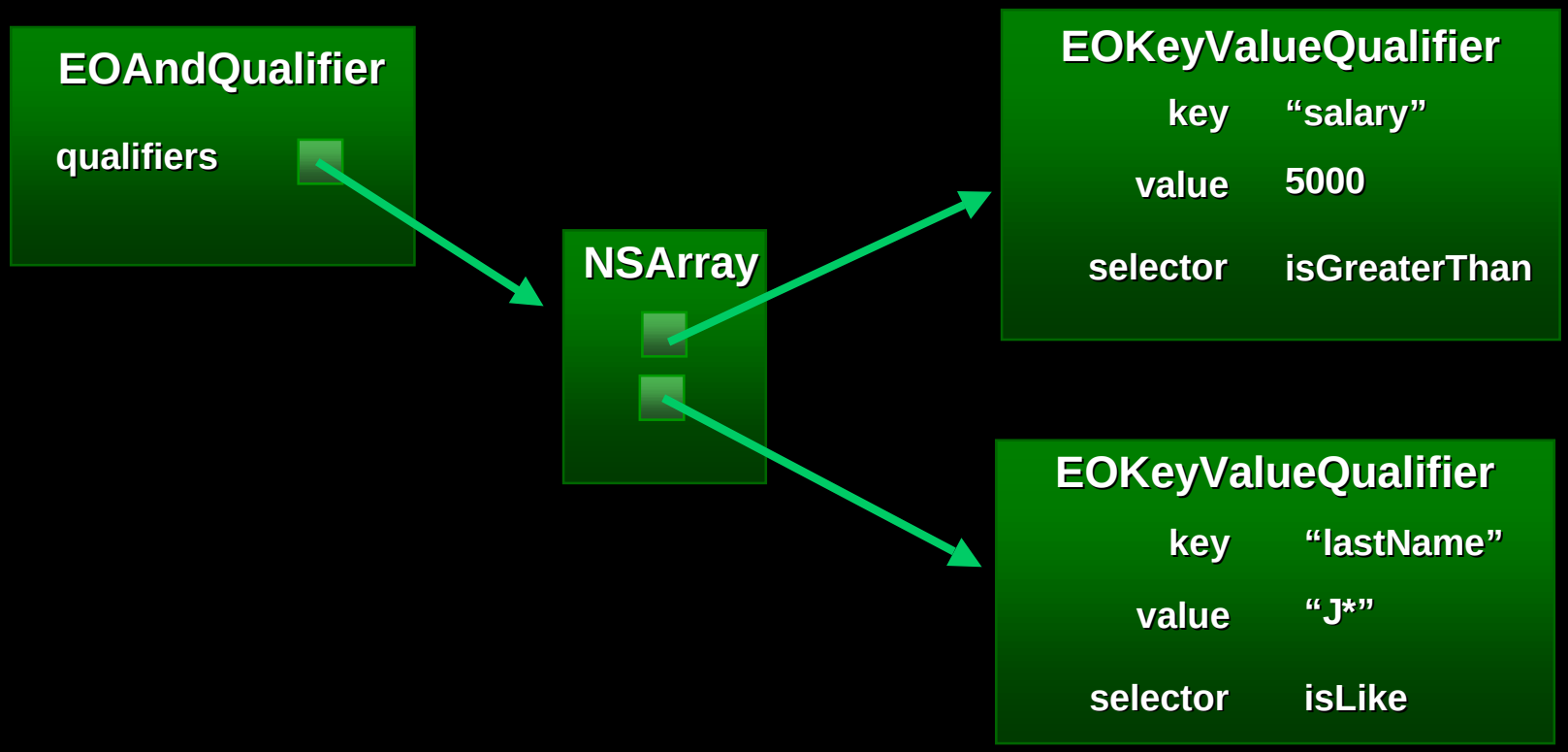
EOFetchSpecification

Describes what should be fetched from the database



EOQualifier

“salary > 5000 AND lastName is like J*”



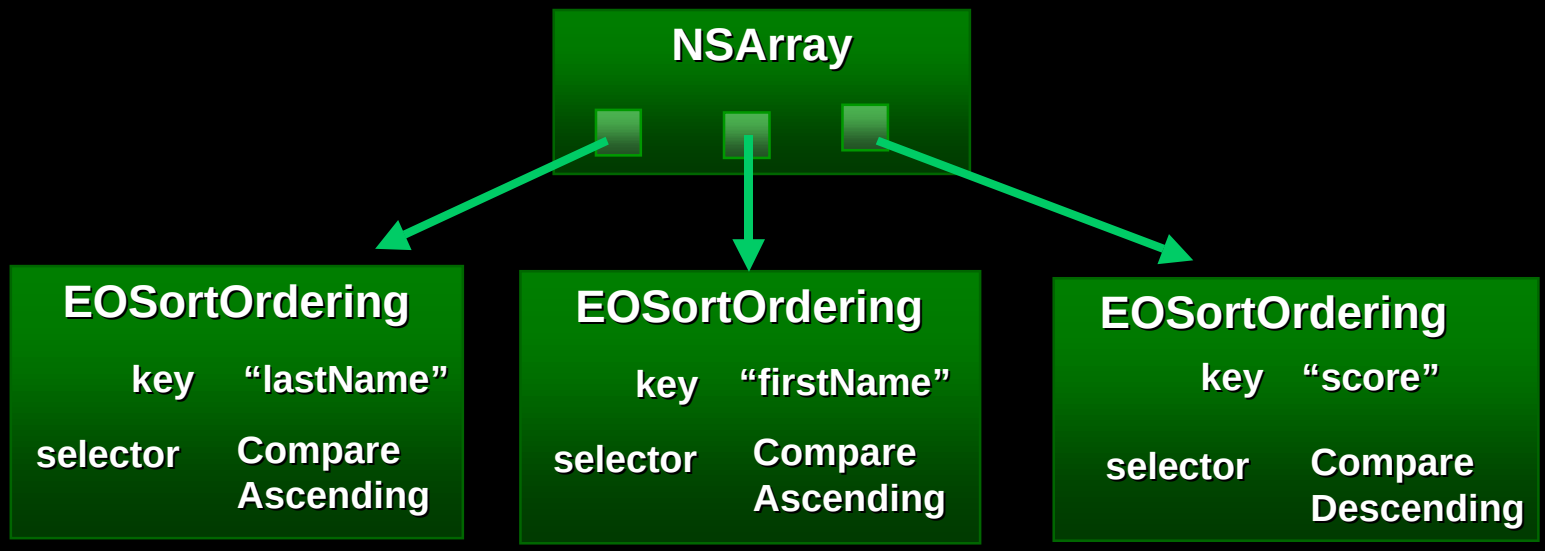
Creating Qualifiers

- Method takes a format string and an array of arguments

```
NSMutableArray args = new NSMutableArray();
args.addObject(new Integer(5000));
args.addObject("J*");

EOQualifier qualifier =
    EOQualifier.qualifierWithQualifiierFormat
    ("salary > %@ AND lastName like %@", args);
```

EOSortOrdering



lastName	firstName	score
Johan	Martin	95
Jones	Allan	90
Jones	Allan	85
Jones	David	87

Creating Sort Orderings

- Sort ordering selectors are static variables of the EOSortOrdering class

```
NSMutableArray orderings = new NSMutableArray();

EOSortOrdering ordering1 =
    new EOSortOrdering("lastName",
        EOSortOrdering.CompareAscending);

orderings.addObject(ordering1);
```

Building the Fetch Spec

- Only the entity name is required for an unqualified, unsorted fetch

```
// assume these have been created
NSMutableArray orderings;
EOQualifier qualifier;

EOFetchSpecification spec =
    new EOFetchSpecification
            ("Employee", qualifier, orderings);
```

Performing the Fetch

- Fetching returns an array of EOs

```
EOFetchSpecification spec;
NSArray results;

EOEditingContext ec =
    session().defaultEditingContext();

results = ec.objectsWithFetchSpecification(spec);
```

Creating and Inserting

- Two separate steps combined in one convenience method

```
EOEnterpriseObject eo;
EOEditingContext ec;

eo =
    EOUtilities.createAndInsertInstance
            (ec, “Employee”);
```

Editing

- Make changes using standard accessors—the editing context observes all changes

```
Employee employee;

employee.setName("Jerry Long");
```

- Key Value Coding provides generic accessors

```
employee.takeValueForKey("Jerry Long", "name");

employee.valueForKey("name");
```

Deleting

- Tell the editing context to delete the object. It will be deleted next time changes are saved

```
EOEnterpriseObject eo;
EOEditingContext ec;

ec.deleteObject(eo);
```

Saving and Reverting Changes

- Pending changes are not sent to the database until you tell editing context to save changes

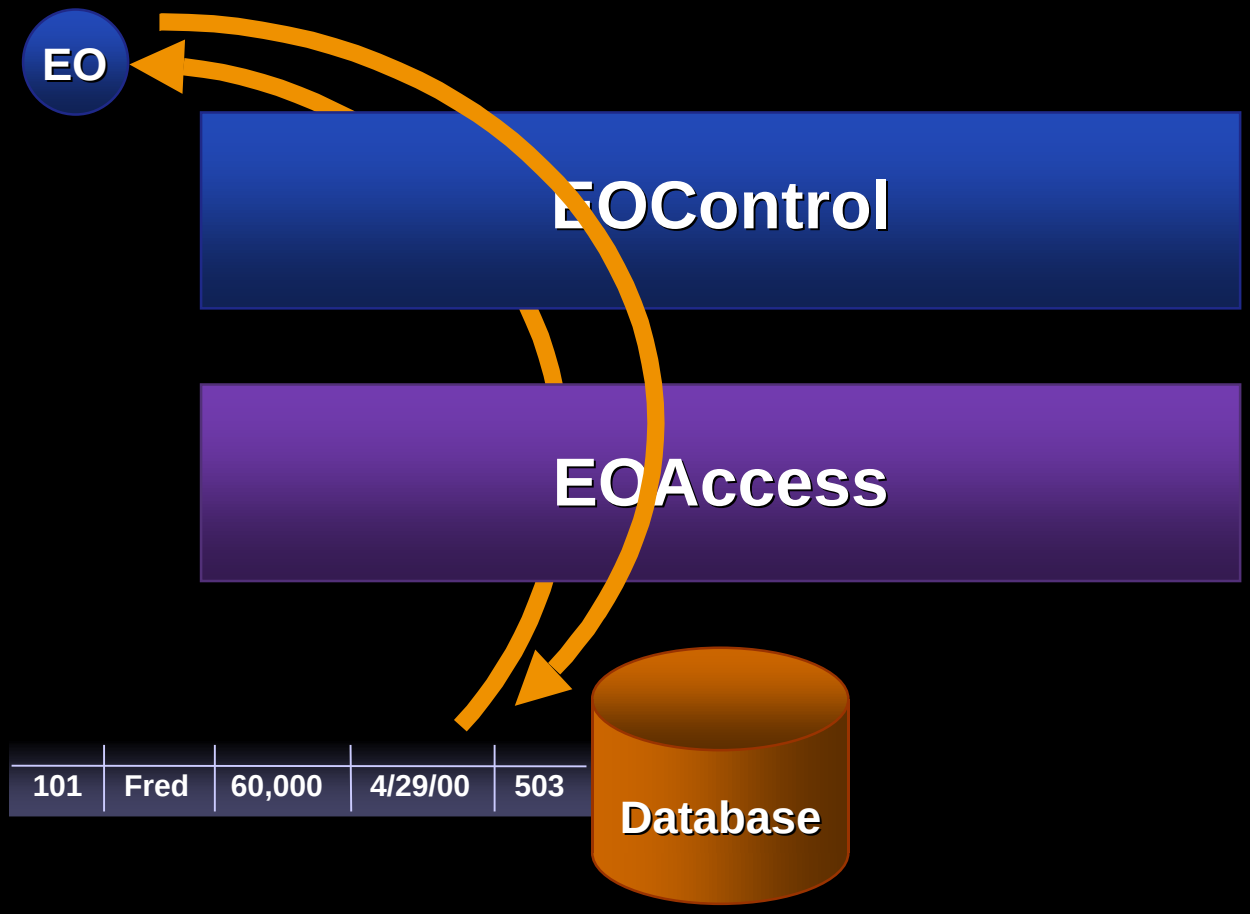
```
editingContext.saveChanges();
```

- Pending changes are discarded by telling the editing context to revert

```
editingContext.revert();
```

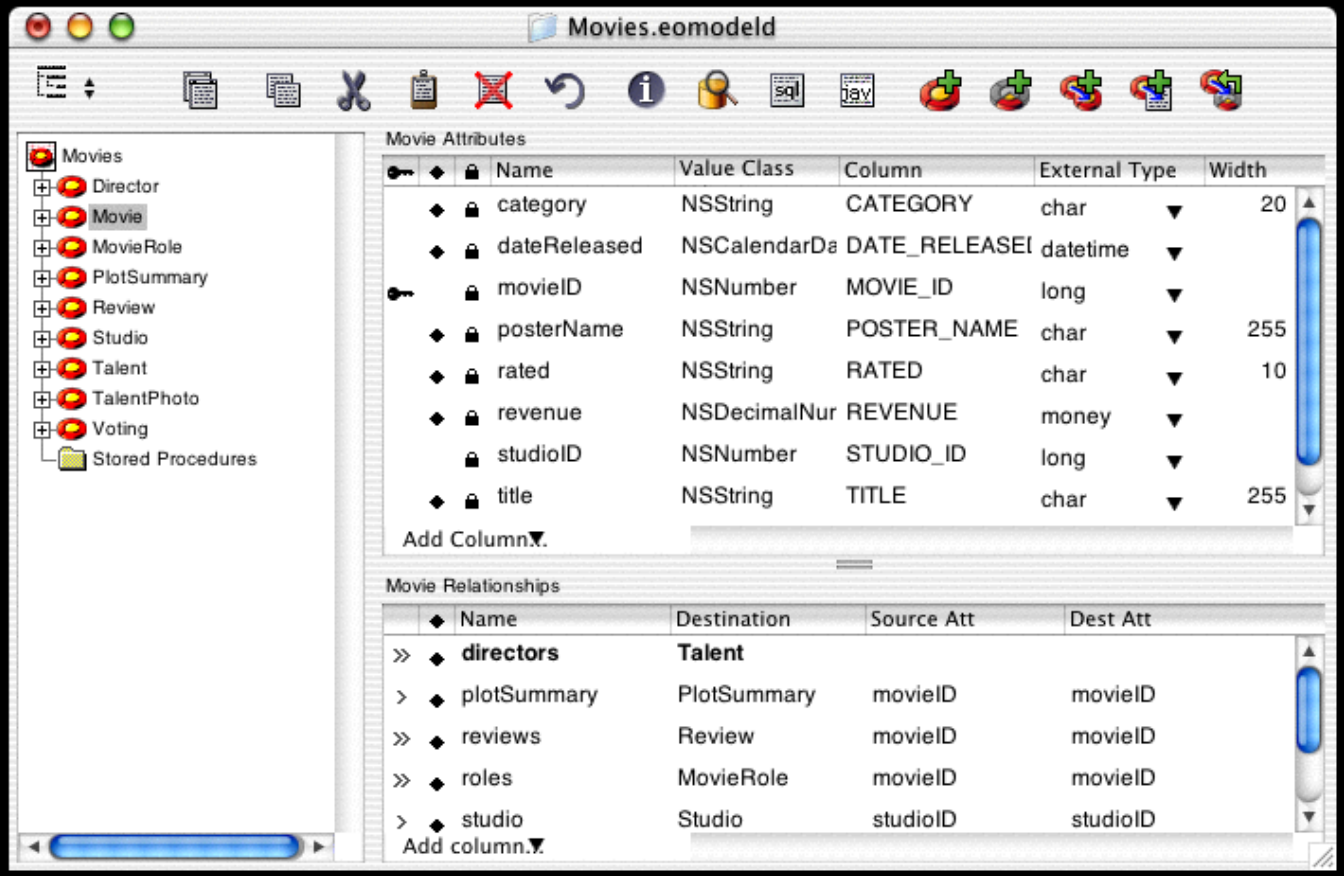
The EOF Stack

Who are the frameworks in your neighborhood?

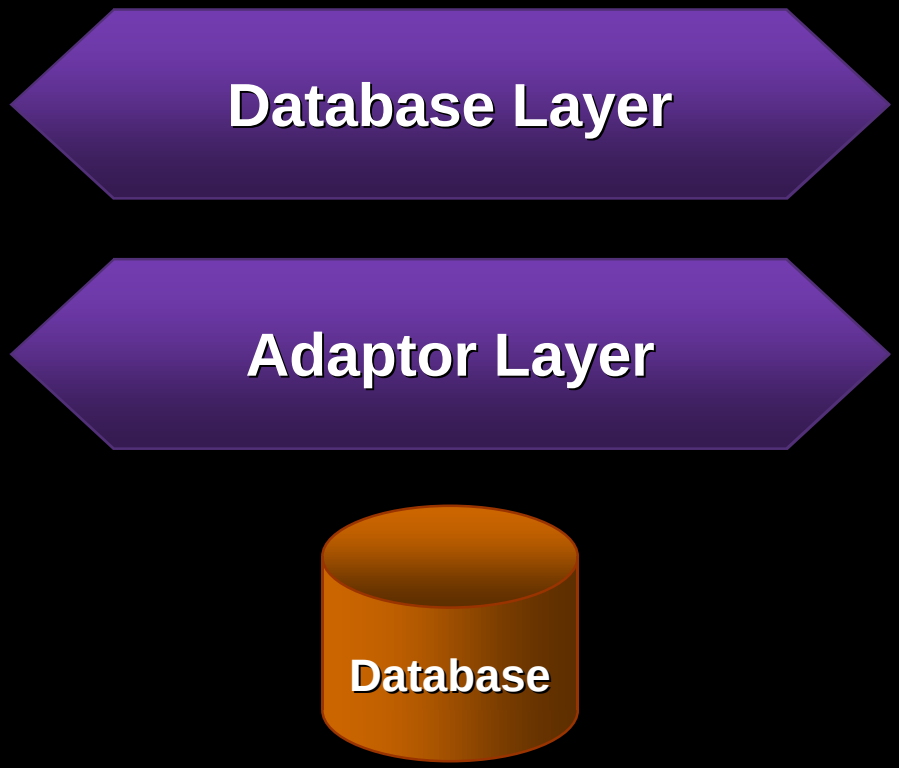


Model Configures EOAccess

Define object-relational mapping in EOModeler



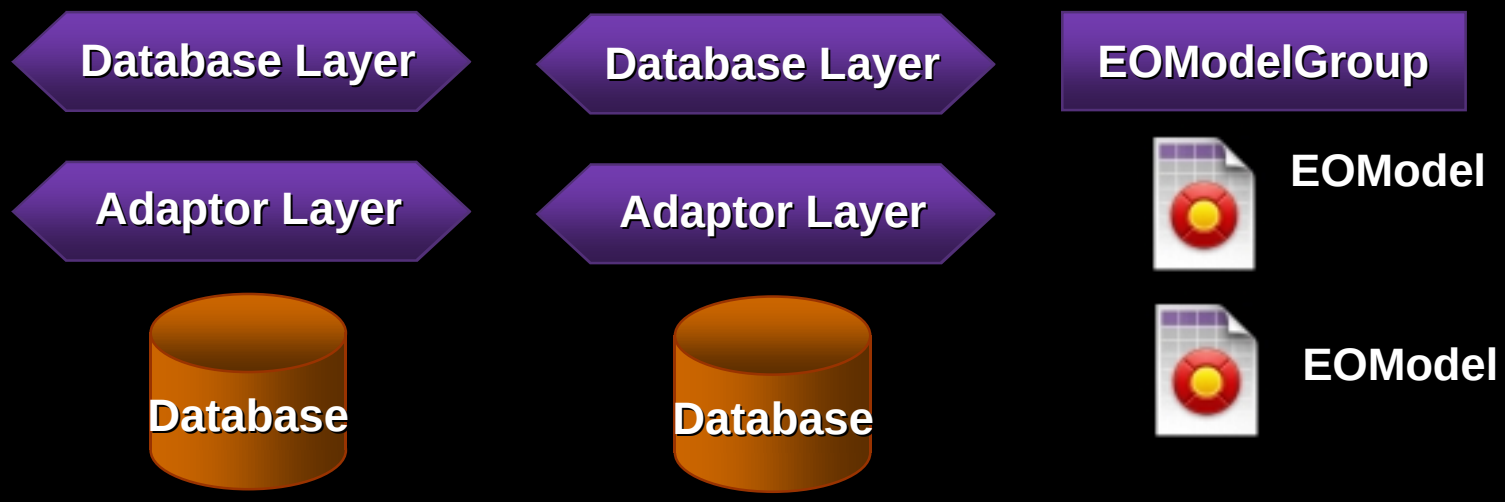
EOAccess Framework



EOModel

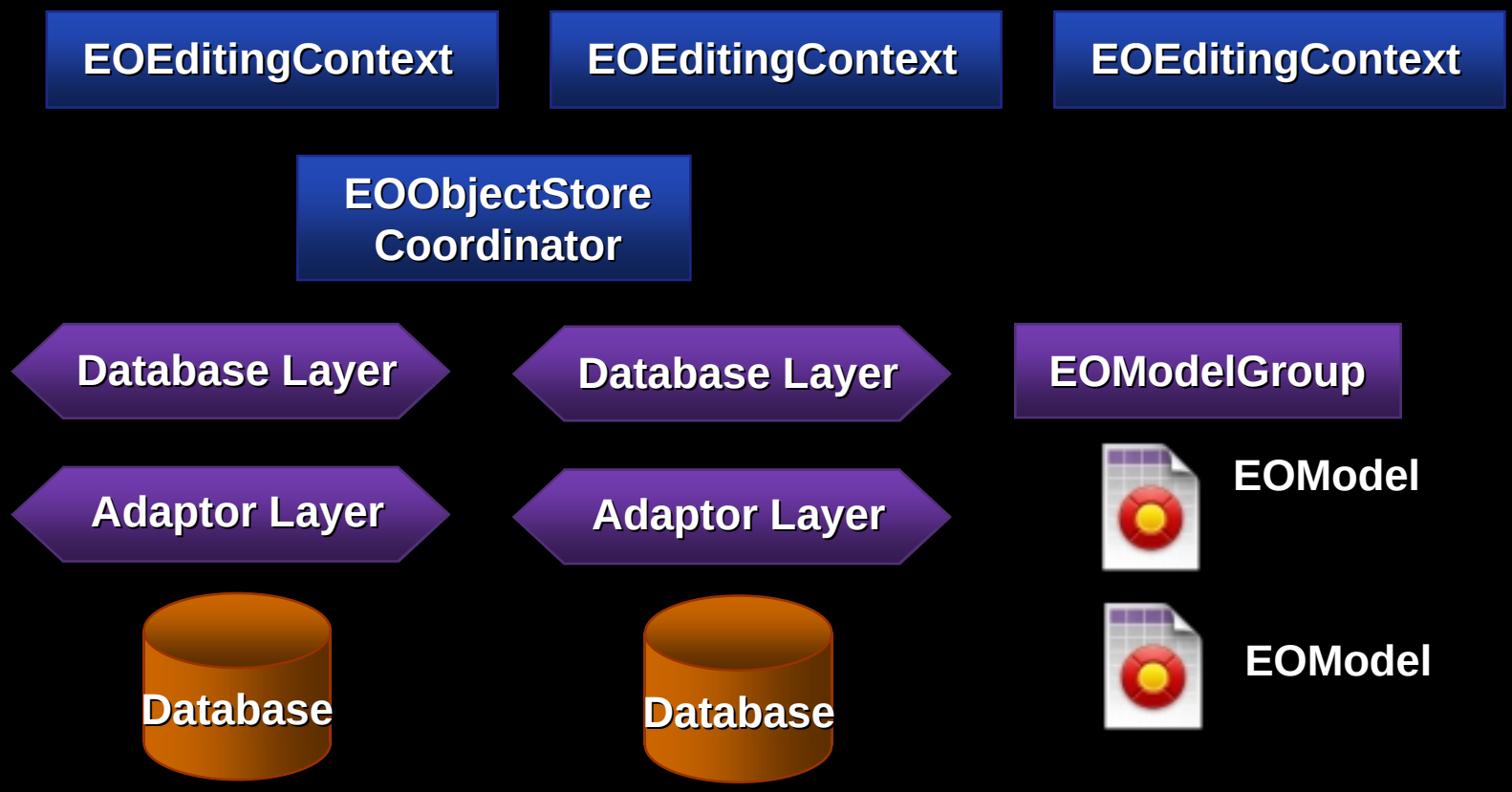
Multiple Data Sources

Multiple EOModels, one per data source

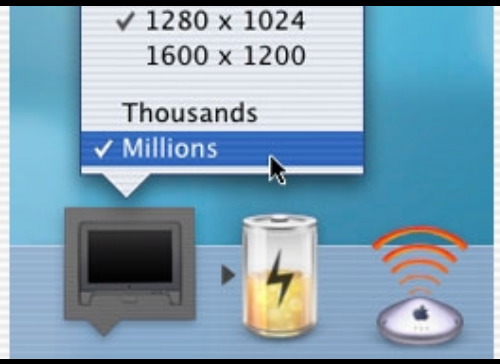


EOControl Framework

EOAccess coordination and programming control



Demo



James Dempsey
Curriculum Manager
Apple iServices Training

WebObjects Lab

- Located downstairs in Room K
- Lab hours
  - Monday 12:00–6:00
  - Tuesday 10:30–6:00
  - Wednesday 10:30–6:00
  - Thursday 10:30–6:00
  - Friday 10:30–3:30

Vote for WebObjects as Best App Server

http://developer.apple.com/webobjects

Roadmap

610 Introduction to Data Modeling & Connectivity	Room A1 **Thurs., 9:00am**
613 Advanced Enterprise Objects Framework	Room A1 **Thurs., 3:30pm**
Session FF111 Feedback Forum: WebObjects	Room A1 **Fri., 3:30pm**

Who to Contact

Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

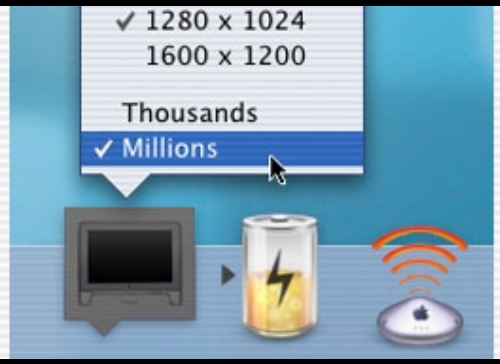
Bob Fraser
WebObjects Product Manager
webobjects@apple.com

iServices Consulting, Integration, Training & Certification
(800) 848-6398
iservices@apple.com

For more information:
www.apple.com/webobjects
developer.apple.com/webobjects
www.apple.com/iservices
www.info.apple.com/webobjects

Subscribe to:
Webobjects-announce@apple.com

Q&A



James Dempsey
Curriculum Manager, iServices Training
iservices@apple.com

http://developer.apple.com/events/urls.html

WWDC

Worldwide Developers Conference 2001